



Gramer ve Diller

Prof.Dr. Banu Diri



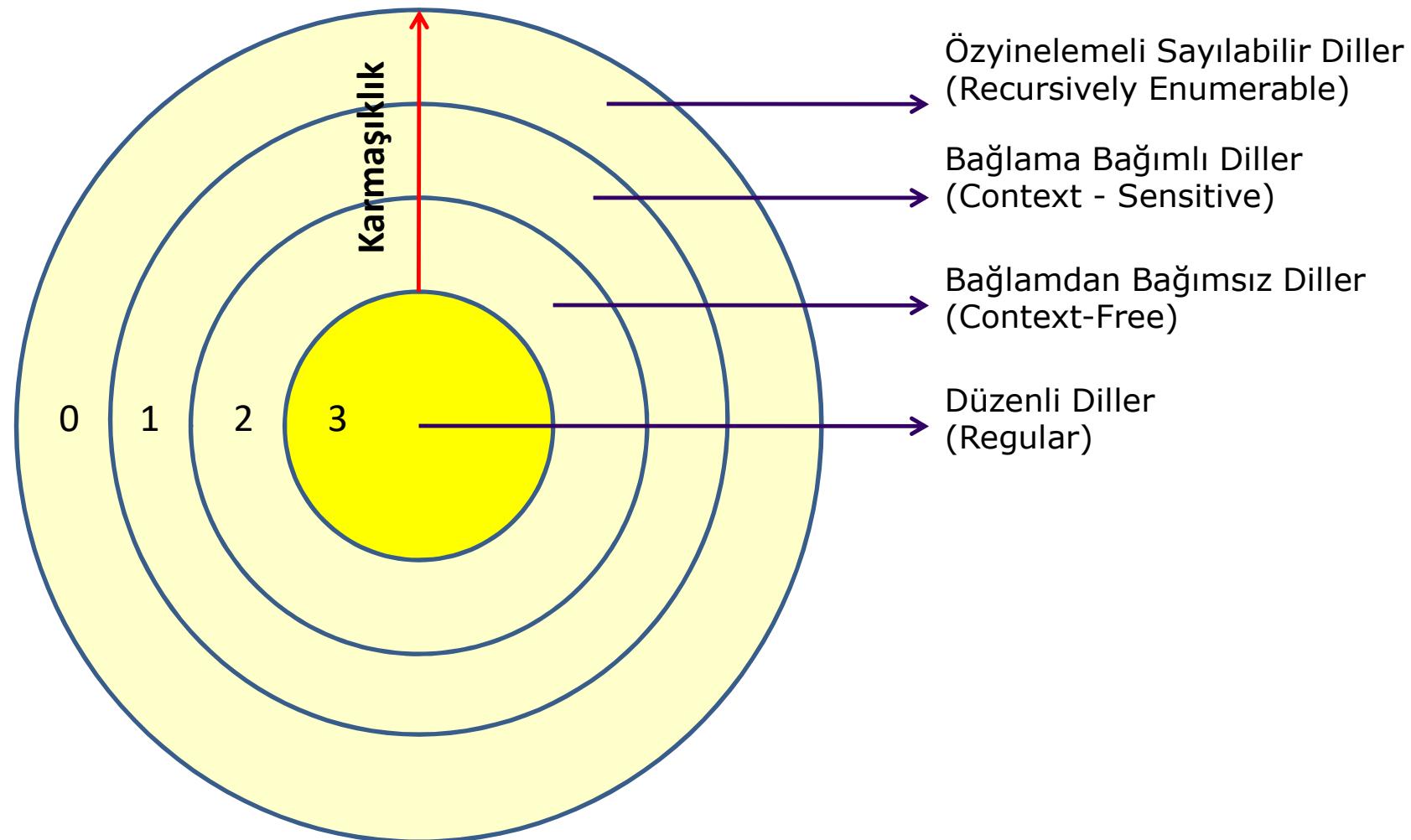
-
-
1. Her biçimsel dil belirli bir alfabe üzerinde tanımlanır.
 2. Alfabe sonlu sayıda simgelerden oluşan bir kümedir.
 3. Alfabeteki simgelerin arka arkaya getirilmesi ile dizgiler (string) oluşturulur.
 4. Biçimsel dil, bir alfabeteki simgelerden oluşturulan dizgilerin bir kümeleridir. Bu kümeyi E ile gösterirsek, bu alfabe üzerinde tanımlanan her dil E 'nin bir alt kümeleridir. E deki her dizgi dilin tümcelerini (sentence) oluşturur.
 5. Bir alfabe üzerinde tanımlanan biçimsel bir dil, bu alfabeteki simgelerden oluşan dizgileri **geçerli** ve **geçersiz** diye ikiye ayırır. Dilde yer alan ve tümceleri oluşturan dizgiler **geçerli** tersi ise **geçersiz**'dir.

-
-
6. Biçimsel dil açısından dizgi, tümce ve sözcük terimleri birbirlerinin yerine kullanılabilir. Tümce dilde yer alan dizgi veya sözcükleri anlatmak için kullanılır. Buna göre, bir alfabe ve bu alfabe üzerinde tanımlı bir dil düşünüldüğünde, alfabeteki simgelerden oluşturulan ve dilde yer alan geçerli dizgiler dilin tümcelerini oluşturmaktadır.
 7. Dilin hangi tümcelerden olduğunu gösteren kurallar bütünü ise dilbilgisi (grammar) olarak adlandırılır.

Büçimsel dilbilgisi ve dillerin incelenmesinde, değişik harf grupları değişik alanlarda kullanılır.

Harf Grubu	Örnekler	Kullanım alanları
Latin alfabetesinin başındaki büyük harfler	A,B,C,...	Sözdizim değişkenleri
Latin alfabetesinin başındaki küçük harfler ve rakamlar	a,b,c,...,0,1,2	Uç simgeler
Latin alfabetesinin sonundaki büyük harfler	U,V,W,Y,...	Sözdizim değişkeni ya da üç simgeler
Latin alfabetesinin sonundaki küçük harfler	u,v,w,y,...	Uç simge dizgileri (sözcükler)
Yunan alfabetesinin başındaki küçük harfler	$\alpha, \beta, \gamma, \dots$	Tümcesel yapılar

Chomsky Hiyerarşisi



Sırasıyla;

- Dilbilgisi ve Dilin Biçimsel Tanımı
- Dilbilgisi ve Dillerin Sınıflandırılması
tür-0, tür-1, tür-2, tür-3
- Sağ-doğrusal ve Sol-doğrusal Dilbilgisi
- Türetme/Ayrıştırma Ağacı
 - Chomsky Normal Biçimi
 - Yukarıdan-aşağıya Ayrıştırma (Top-down parsing)
 - Aşağıdan-yukarıya Ayrıştırma (Bottom-up parsing)

**tür-0, tür-1, tür-2, tür-3
Örnekler ile anlatılacaktır**

Dilbilgisi ve Dilin Tanımı

BİGİMSİZ olarak dilbilgisi bir dörtlü olarak tanımlanır.

$$G = \langle V_N, V_T, P, S \rangle$$

V_N : Söz dizim değişkenleri kümesi (sonlu bir kümeye)

V_T : Uç simgeler kümesi (sonlu bir kümeye)

V_N ve V_T ayırik kümelerdir. $V_N \cap V_T = \emptyset$

S : Başlangıç değişkeni $S \in V_N$

P : Yeniden yazma ve türetme kuralları

$\alpha \Rightarrow \beta$ (α 'nın yerine β konulabilir)

En genel bİGİMSİZle

$$\alpha \in V^+ \quad \beta \in V^*$$

$$V = V_N \cup V_T \quad V^+ = V^* - \{\lambda\}$$

Bir dilbilgisi tarafından tanımlanan dil bireimsel olarak aşağıdaki gibi tanımlanır.

$$L(G) = \{ w \mid w \in V_T^*, S \xrightarrow{*} w \}$$

Bu tanımına göre, bir dilin tümçeleri, başlangıç simgesinden (S' den) başlanarak ve yeniden yazma kuralları yeterli sayıda kullanılarak elde edilen üç simge dizgeleridir.

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow w$$

$\alpha_1, \alpha_2, \dots, \alpha_n$: tümcesel yapılar

w: tümce

ÖRNEK $G_1 = \langle V_N, V_T, P, S \rangle$

$$V_N = \{ S \}$$

G_1 tarafından türetilen bir kagıt tümce

$$V_T = \{ 0, 1 \}$$

$$S \Rightarrow 01$$

$$P: \quad S \Rightarrow 0S1$$

$$S \Rightarrow 0S1 \Rightarrow 0011$$

$$S \Rightarrow 0S1$$

$$\Rightarrow 00S11 \Rightarrow 000111$$

Bu örneklerden yararlanarak dili

$$L(G_1) = \{ 0^n 1^n \mid n \geq 1 \}$$

Dilbilgisi ve Dillerin Sınıflandırılması

Dilbilgisi ve türetilkleri diller, yeniden yazma kurallarının özellikleri -
ne göre:

- tür-∅ (kısıtlamasız dilbilgisi ve diller)
- tür-1 (bağlama - bağımlı dilbilgisi ve diller)
- tür-2 (bağlamdan - bağımsız dilbilgisi ve diller) (Context Free Grammars)
- tür-3 (düzenli dilbilgisi ve diller)

Tür-∅ Dilbilgisi ve Dil

Tür-∅'in yeniden yazma kuralları

$$\alpha \Rightarrow \beta : \alpha \in V^+ \quad \beta \in V^*$$

Bu tip dillere özyinelemeli sayılabilir (recursively enumerable) dillerde
denir. Dilin tümelerini ard arda türeten bir yordam vardır.

ÖRNEK

$$G_2 = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S, L, R, A, B, C \}$$

$$V_T = \{ a \}$$

aaaa

$$P: S \Rightarrow LAaR$$

$$Aa \Rightarrow aaA$$

$$AR \Rightarrow BR | CR$$

$$aB \Rightarrow Ba$$

$$LB \Rightarrow LA$$

$$aC \Rightarrow Ca$$

$$LC \Rightarrow \lambda$$

$$S \Rightarrow LA\underline{a}R \Rightarrow L\underline{aa}AR \Rightarrow La\underline{aa}C \Rightarrow La\underline{C}a \Rightarrow \underline{L}Caa \Rightarrow aa$$

$$\begin{aligned} S &\Rightarrow LA\underline{a}R \Rightarrow L\underline{aa}AR \Rightarrow La\underline{aa}B \Rightarrow La\underline{B}aR \Rightarrow \\ &\quad \underline{L}BaR \Rightarrow La\underline{aa}R \Rightarrow La\underline{aa}A \Rightarrow \\ &\quad La\underline{aaa}AR \Rightarrow La\underline{aaa}C \Rightarrow La\underline{aa}Ca \\ &\quad La\underline{a}Ca \Rightarrow La\underline{C}aa \Rightarrow La\underline{C}aaaa \Rightarrow aaaa \end{aligned}$$

Dilbilgisinin kuralları dikkatle incelendiğinde $L(G_2)$ dilinin tanımı

$$L(G_2) = \{ a^i \mid i = 2^n, n \geq 1 \}$$

Tür-1 Dilbilgisi ve Dil

Tür-1'in yeniden yazma kuralları:

$$\alpha \Rightarrow \beta : \alpha \in V^+ \quad \beta \in V^* \quad |\alpha| \leq |\beta|$$

Tür-1'de, Tür-0'a göre $|\alpha| \leq |\beta|$ kısıtlaması getirilmiştir.

Tür-1 dilbilgisine bağlama- bağımlı (context sensitive) de denir.
veya özyineli (r.)

Günlük tür-1 dilbilgisi, yeniden yazma kurallarının tümü

$\alpha_1 A \alpha_2 \Rightarrow \alpha_1 B \alpha_2 \quad A \in V_N, \alpha_1, \alpha_2, \beta \in V^*$ olan normal bir forma dönüştürülebilir. Bu normal biçimde $\alpha_1 \dots \alpha_2$ bağlamında A'nın yerine β konulabildiği için, dilbilgisi bağlama bağımlı dilbilgisi olarak adlandırılır.

ÜRNEK

$G_3 \langle V_N, V_T, P, S \rangle$

$$V_N = \{ S, A, B \}$$

$$V_T = \{ a, b, c \}$$

$$S \Rightarrow a\underline{A}B \Rightarrow ab\underline{B} \Rightarrow abc$$

$$S \Rightarrow a\underline{SAB} \Rightarrow aa\underline{ABAB} \Rightarrow aa\underline{bB}\underline{AB} \Rightarrow$$

$$aa\underline{ab}\underline{AB}\underline{B} \Rightarrow a\underline{abb}\underline{BB} \Rightarrow a\underline{bb}\underline{bc}\underline{B}$$

$$\Rightarrow a\underline{abb'cc}$$

$$P: \quad S \Rightarrow aSAB$$

$$S \Rightarrow aAB$$

$$BA \Rightarrow AB$$

$$aA \Rightarrow ab$$

$$bA \Rightarrow bb$$

$$bB \Rightarrow bc$$

$$cB \Rightarrow cc$$

$$L(G_3) = \{ a^n b^n c^n \mid n \geq 1 \}$$

Tür-2 Dilbilgisi ve Dil

Tür-2 dilbilgisinin yeniden yazma kuralları

$A \Rightarrow^* B : A \in V_N \quad B \in V^*$ biçimindedir.

Yeniden yazma kurallarının sol tarafında tek bir değişken (A) yer almaktadır. Yeniden yazma kuralı, hangi bağlamda olursa, A 'nın yerine B konulabileceğini söyleyelim. Bu dilbilgi-sine Bağlamdan-Bağımsız (context-free) dilbilgisi denir.

Programlama dilleri Bağlamdan-Bağımsız dilbilgisidir

DRNEK

$$G_1 = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

$$V_T = \{+, -, *, /, (,), v, c\}$$

$$P: S \Rightarrow S+S \mid S*S \mid S-S \mid S/S \mid (S) \mid v \mid c$$

$$\begin{aligned} S &\Rightarrow S * \underline{S} \Rightarrow S * (\underline{S}) \Rightarrow \underline{S} * (S-S) \Rightarrow v * (\underline{S}-S) \Rightarrow v * (v-S) \\ &\Rightarrow v * (v-c) \end{aligned}$$

Tür-3 Dilbilgisi ve Dil

Tür-3 dilbilgisinin yeniden yazma kuralları

$$A \Rightarrow aB$$

$$A \Rightarrow a$$

$$A \Rightarrow \lambda : A, B \in V_N \quad a \in V_T \quad \text{biçimindedir.}$$

Yeniden yazma kurallının sol taraflında tek bir değişken (A), sağ taraflında ise ya tek bir uç simge ya da bir uç simge ile bir değişken yer almaktadır. Bu dilbilgisine Düzgün (regular) dil denir.

ÖRNEK

$$G_5 = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S, A, B\}$$

$$V_T = \{0, 1\}$$

$$P: S \Rightarrow OS | OA | OI | \lambda$$

$$A \Rightarrow OB$$

$$B \Rightarrow IS$$

$$S \Rightarrow OS \Rightarrow OO$$

$$S \Rightarrow OA \Rightarrow OO \underline{B} \Rightarrow OOIS \Rightarrow OOI$$

$$S \Rightarrow OS \Rightarrow OO \underline{A} \Rightarrow OOO \underline{B} \Rightarrow OOOIS$$

$$\Rightarrow OOO1O \underline{A} \Rightarrow OOO1OO \underline{B} \Rightarrow$$

$$OOO1OOIS \Rightarrow OOO1OO10$$

$L(G_5) \rightarrow$ içerişindeki her 1'den önce en az iki tane 0 bulunan dizgedir.

$$(\emptyset)^* (001 (\emptyset)^*)^n (\emptyset)^* \quad n \geq 0$$

$$((00)^+ 1)^* \emptyset^*$$

Sağ Doğrusal – Sol Doğrusal Dilbilgisi

Yeniden yazma kuralları

$$A \Rightarrow wB$$

$$A \Rightarrow w = A, B \in V_N, w \in V_T^*$$

biriminde olan dilbilgisine sağ-doğrusal (right-linear) dilbilgisi denir. Sağ-doğrusal dilbilgisinin yeniden yazma kurallarının sol tarafında bir değişken, sağ tarafında ise bir uç simgeler dizgesi veya bir uç ^{şimgeler} dizgesi ile bir değişken yer alır. Uç simgeler dizgesi sıfır uzunlığunda bir dizide olabilir.

Yeniden yazma kuralları

$$A \Rightarrow Bw$$

$A \Rightarrow w = A, B \in V_N, w \in V_T^*$ biçiminde olan dilbilgisine sol-doğrusal (left-linear) dilbilgisi denir. Sol-doğrusal dilbilgisinin yeniden yazma kurallarının sol tarafında bir değişken, sağ tarafında ise bir uç simgeler dizgesi veya bir değişken ile bir uç simgeler dizgesi yer alır. Uç simgeler dizgesi sıfır uzunlığunda bir dizide olabilir.

Sağ-dogrusal ve sol-dogrusal dilbilgileri tarafından türetilen diller düzgün dillerdir. Tür-3 dilbilgileri tarafından türetilen dillerde düzgün diller olduğuna göre tür-3, sağ-dogrusal ve sol-dogrusal dilbilgileri denk dilbilgileridir.

ÖRNEK $G_6 = \langle V_N, V_T, P, S \rangle$ Sağ doğrusal bir dilbilgisidir.

$$V_N = \{A, B, S\}$$

$$V_T = \{0, 1\}$$

$$\begin{aligned} P: \quad & S \Rightarrow 0A \\ & A \Rightarrow 10A \mid \lambda \end{aligned}$$

$$S \Rightarrow 0\underline{A} \Rightarrow 0$$

$$S \Rightarrow 0\underline{A} \Rightarrow 01\underline{A} \Rightarrow 010$$

$$\begin{aligned} S \Rightarrow 0\underline{A} \Rightarrow 010\underline{A} \Rightarrow 01010\underline{A} \\ \Rightarrow 01010 \end{aligned}$$

$$L(G_6) = 0(10)^* = (01)^*0$$

Türetme Ağacı

Bir dilbilgisi tarafından türetilen tümcesel yapı ve tümceler su şekilde gösteriliyor olsun:

$$S \Rightarrow \mathcal{B}_1 \Rightarrow \mathcal{B}_2 \Rightarrow \mathcal{B}_3 \dots \Rightarrow \mathcal{B}_{n-1} \Rightarrow \mathcal{B}_n \Rightarrow \underbrace{w}_{\text{Tümce}}$$

\swarrow
Tümcesel yapılar

$$\mathcal{B}_i \in V^*, w \in V_T^*$$

Bağlamdan - bağımsız dillerin her tümcesel yapısına ve her tümcesine türetme veya ayrıştırma ağacı (derivation or parsing tree) karşı gelir.

ÖRNEK

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S, A\}$$

$$V_T = \{0, 1\}$$

$$P: S \Rightarrow \emptyset S \emptyset \mid 1 S 1 \mid \emptyset A \emptyset \mid 1 A 1$$

$$A \Rightarrow \emptyset A 1 \mid 1 A \emptyset \mid \emptyset 1 \mid 1 \emptyset$$

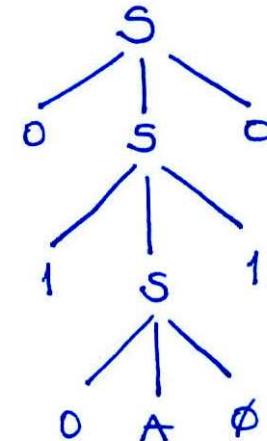
türetilen

$\beta = 010A010$ tümcesel yapısı ile

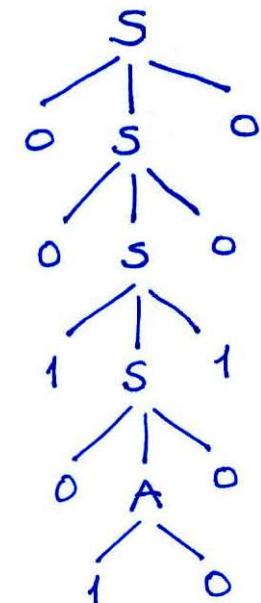
$w = 0010100100$ tümcesine

karşı gelen türetme ağacıları

$\beta \Rightarrow$



$w \Rightarrow$



Türetme ve Ayrıştırma Ağacının Tanımı

- 1) Ağacın kökünün etiketi S' dir.
- 2) Kök dışındaki ara düğümlerin etiketleri sözdizim değişkenleridir ($A \in V_N$)
- 3) Eğer ağaç bir tümcesel yapıya karşı geliyorsa, yaprakların etiketleri sözdizim değişkenleri veya uç simgeler olabilir ($X \in V$). Eğer ağaç bir tümceye karşı geliyorsa yaprakların etiketleri sadece uç simgelerdir ($a \in V_T$) olabilir.
- 4) Eğer bir ara düğümün etiketi A , bu ara düğümün hemen altındaki düğümlerin etiketleri soldan sağa x_1, x_2, \dots, x_k ise, dilbilgisinin yeniden yazma kuralları arasında
$$A \Rightarrow x_1 x_2 x_3 \dots x_k \quad x_1, x_2, x_3, \dots, x_k \in V$$
kuralı yer almmalıdır.
- 5) Eğer bir düğümün etiketi λ ise, bu düğüm bir uç düğüm (yaprak) olmalı ve bu düğümün kardeşi bulunmalıdır.

Soldan ve Sağdan Türetme

Yapraklarının etiketleri üç simgeler olan her türetme ağacına dilin bir tümcesi karşı gelir. Dil belirgin (unambiguous) bir dil ise, dildeki her tümceye bir türetme ağacı gelir, eğer bir tümceye birden fazla türetme ağacı karşı geliyorsa dil belirgin olmayan (ambiguous) bir dildir.

X } Tümceleri birden çok anlam taşıyan belirgin olmayan
dillerin uygulamada bir değeri yoktur.

Ex: I saw the man in the park with a telescope

Eğer bir türmce türetilirken, her adımda en soldaki değişkene bir türetme uygulanıyorsa, yapılan türetmeye soldan türetme (leftmost derivation) denir. Soldan türetmenin her adımda, eğer türmcesel yapı birden çok değişken içeriyorsa, öncelik en soldaki değişkene verilir.

ÖRNEK

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

$$V_T = \{+, -, *, /, (,), v, c\}$$

$$P: S \Rightarrow S+S \mid S-S \mid S*S \mid S/S \mid (S) \mid v \mid c$$

$$w_1 = (v+c) * (v-c)$$

$$w_2 = v/(v-c) + v*(v+c)$$

DİREK

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

$$V_T = \{+, -, *, /, (,), v, c\}$$

$$P: S \Rightarrow S+S \mid S-S \mid S*S \mid S/S \mid (S) \mid v \mid c$$

$$w_1 = (v+c) * (v-c)$$

$$w_2 = v/(v-c) + v*(v+c)$$

w_1 'in soldan türetilmesi

$$\begin{aligned} S &\Rightarrow S*S \Rightarrow (S)*S \Rightarrow (S+S)*S \Rightarrow (v+s)*S \Rightarrow (v+c)*S \\ &\Rightarrow (v+c)*(S) \Rightarrow (v+c)*(S-S) \Rightarrow (v+c)*(v-s) \Rightarrow (v+c)*(v-c) \end{aligned}$$

w_2 'in soldan türetilmesi

$$\begin{aligned} S &\Rightarrow S+S \Rightarrow S/S+S \Rightarrow v/(S)+S \Rightarrow v/(S-S)+S \Rightarrow v/(v-s)+S \\ &\Rightarrow v/(v-c)+S \Rightarrow v/(v-c)+S*S \Rightarrow v/(v-c)+v*S \\ &\Rightarrow v/(v-c)+v*(S) \Rightarrow v/(v-c)+v*(S+S) \Rightarrow v/(v-c)+v*(v+s) \\ &\Rightarrow v/(v-c)+v(v+c) \end{aligned}$$

Eğer bir türme türetilirken, her adımda en sağdaki değişkene bir türetme uygulanıyorsa, yapılan türetmeye sağdan türetme (rightmost derivation) denir. Sağdan türetmenin her bir adımda, eğer türmeseel yapı binden çok değişken içeriyorsa, öncelik en sağdaki değişkene verilir.

DİREK

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

$$V_T = \{+, -, *, /, (,), v, c\}$$

$$P: S \Rightarrow S+S \mid S-S \mid S*S \mid S/S \mid (S) \mid v \mid c$$

$$\omega_1 = (v+c) * (v-c)$$

$$\omega_2 = v/(v-c) + v * (v+c)$$

ω_1 'in sağdan türetilmesi

$$\omega_1 = (v+c) * (v-c)$$

$$S \Rightarrow S*S \Rightarrow S*(S) \Rightarrow S*(S-S) \Rightarrow S*(0-c) \Rightarrow S*(v-c)$$

$$\Rightarrow (S)*(v-c) \Rightarrow (S+S)*(v-c) \Rightarrow (S+v)*(v-c) \Rightarrow (v+c)*(v-c)$$

ω_2 'nin sağdan türetilmesi

$$\omega_2 = v/(v-c) + v * (v+c)$$

$$S \Rightarrow S+S \Rightarrow S + S*S \Rightarrow S + S*S \Rightarrow S + S*(S+S)$$

$$\Rightarrow S + S*(S+v) \Rightarrow S + S*(v+c) \Rightarrow S + v*(v+c)$$

$$\Rightarrow S/S + v*(v+c) \Rightarrow S/(S) + v*(v+c)$$

$$\Rightarrow S/(S-S) + v*(v+c) \Rightarrow S/(S-c) + v*(v+c)$$

$$\Rightarrow S/(v-c) + v*(v+c) \Rightarrow v/(v-c) + v*(v+c)$$

Aşağıdan Yukarıya Doğru Ayırıştırma (Bottom-Up Parsing)

Aşağıdan - yukarıya ayrışmada, ayırtılacak tüm ceden (ω) başlanarak bir dizi işlem sonunda dilbilgisinin başlangıç simgesi olan S elde edilmeye çalışılır. Bu ayırtımada sağdan öncelikli türetme (rightmost derivation) kullanılır.

O'NEK

$$A \Rightarrow abA|ab$$

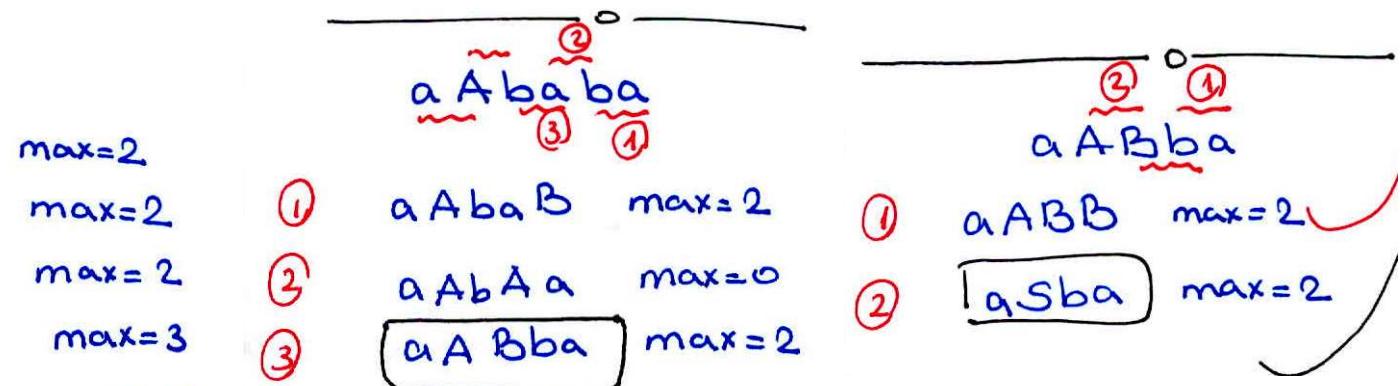
$$S \Rightarrow aS|AB|B$$

$$B \Rightarrow BB|ba$$

"aababbaba"

Sırasıyla 9, 8, 7, 6, 5, 4, 3'ü eşlemelere sağdan ilerliyerek batılır.

a	ab	b	b	a	b	b	a	b	a
①	②	③	④	⑤	⑥	⑦	⑧	⑨	
① aababbab B	max=2	① aAbab B	max=2	① aAB B	max=2	① aAS B	max=2		
② aababbab A	max=2	② aAbab A	max=0	② aAB B	max=2	② aSba	max=2		
③ aababBba	max=2	③ aAbAa	max=0	③ aABba	max=2				
④ aabA bab a	max=3	④ aA Bba	max=2						
⑤ aaBbbaba	max=2								
⑥ aAabbaba	max=2								



a a b a b b a b a

(5) (2)
(6) (4) (3) (1)

- | | |
|---|-------|
| ① a a b a b b a B | max=2 |
| ② a a b a b b A a | max=2 |
| ③ a a b a b B b a | max=2 |
| ④ a a b A b a b a | max=3 |
| ⑤ a a B b b a b a | max=2 |
| ⑥ a A a b b a b a | max=2 |

a A b a b a

(2)
(3) (1)

- | | |
|---|-------|
| ① a A b a B | max=2 |
| ② a A b A a | max=0 |
| ③ a A B b a | max=2 |

a A B b a

(3) (1)

- | | |
|---|-------|
| ① a A B B | max=2 |
| ② a S b a | max=2 |

a S b a

(2) (1)

1 a S B max=2

2 S b a max=2

S b a

S B çözüm yok

a S B

S B çözüm yok

a A B B

(1)

1 a A B max=2
3 a S B max=2

S B çözüm yok

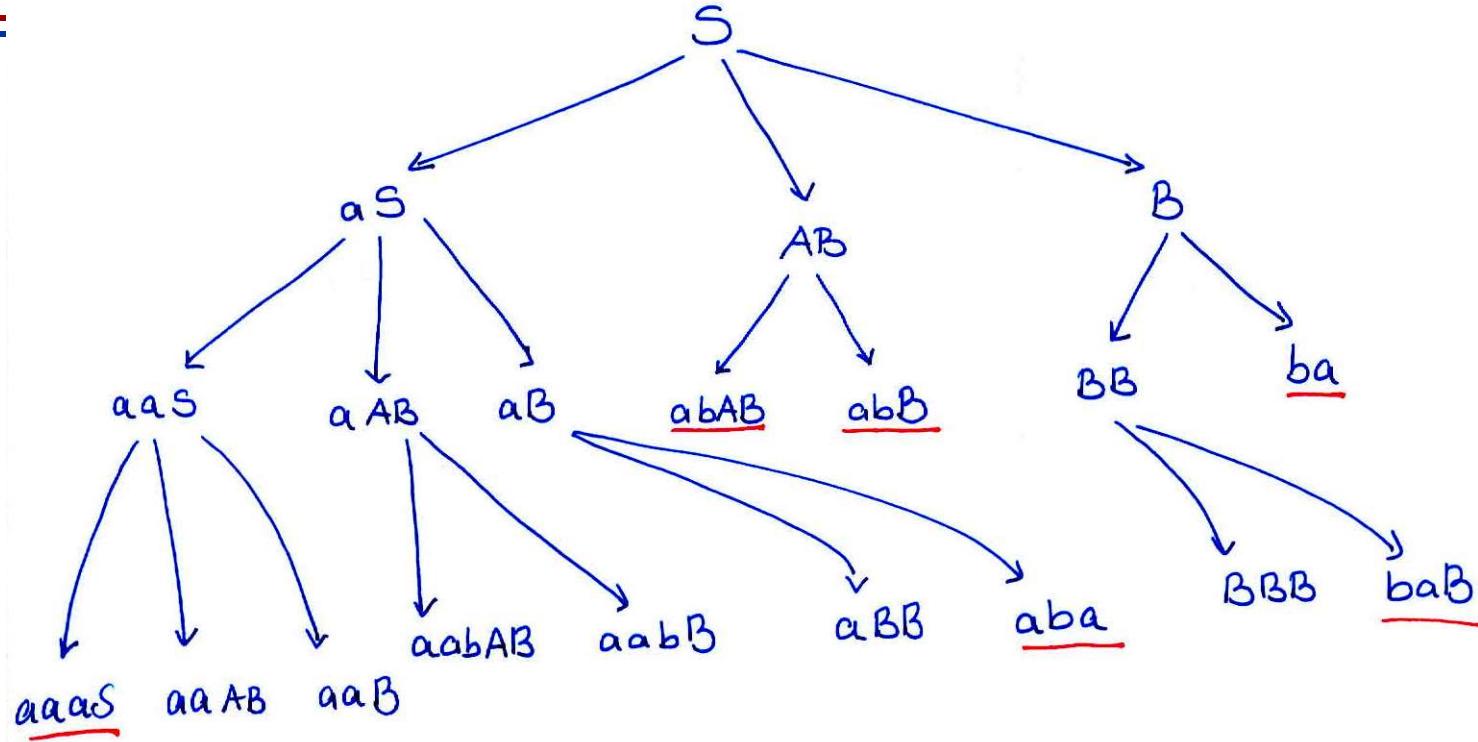
a S \Rightarrow S

Yukarıdan Aşağıya Ayırıştırma (Top-Down Parsing)

Yukarıdan aşağıya ayrıştırmada, dilbilgisinin başlangıç simgesi olan S' den başlanarak, ayırtılacak tümce(w) türetilmeye çalışılır. w türetilmeye çalışılırken soldan türetme (leftmost derivation) kuralları uygulanır.

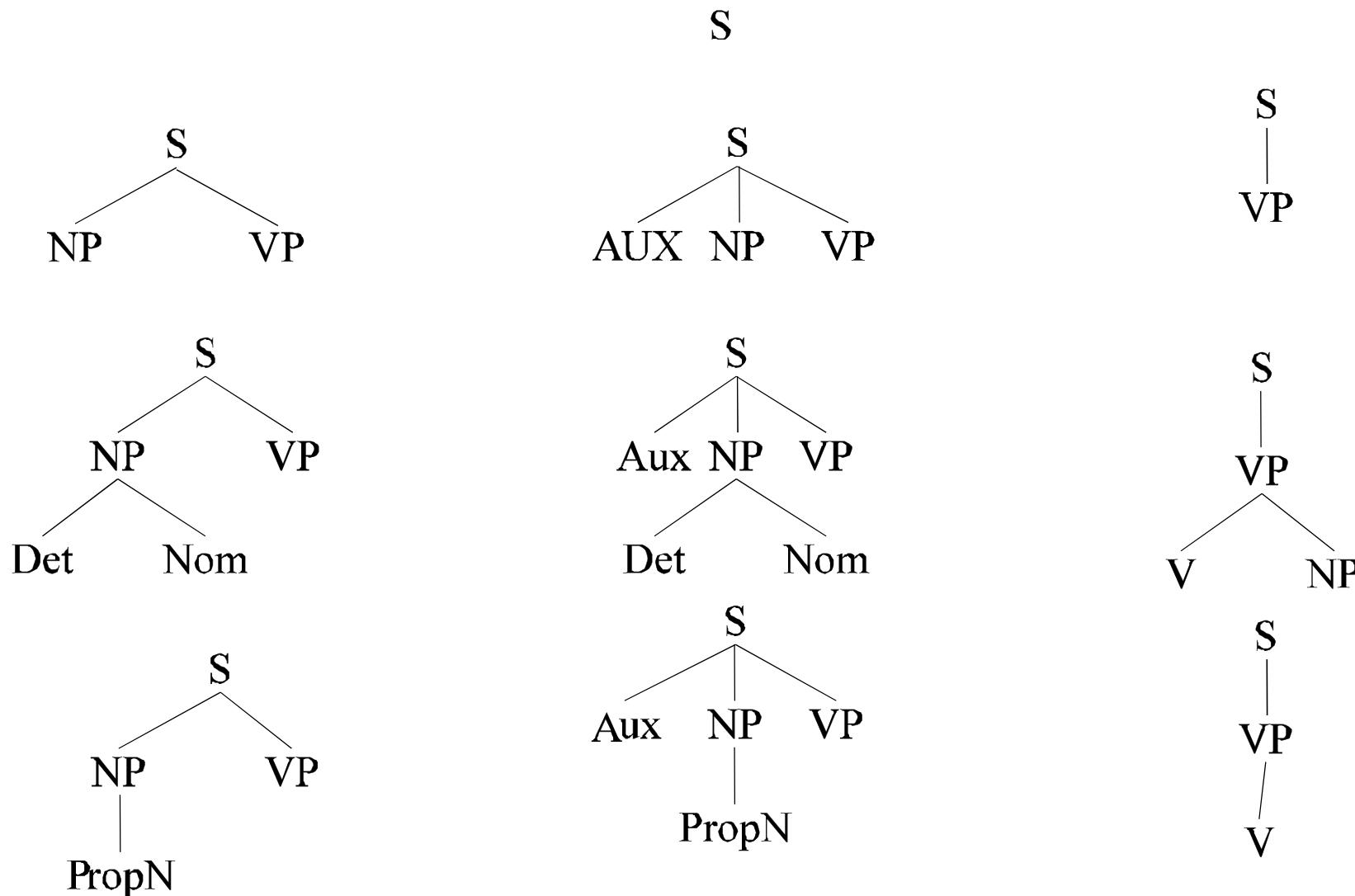
Soldan türetmede, türetmenin herhangi bir adımdındaki tümcesel yapı α ise ve α içindeki sözdizim değişkenlerinden en soldaki A ise, A kurallarından biri uygulanarak yeni bir tümcesel yapı elde edilir.

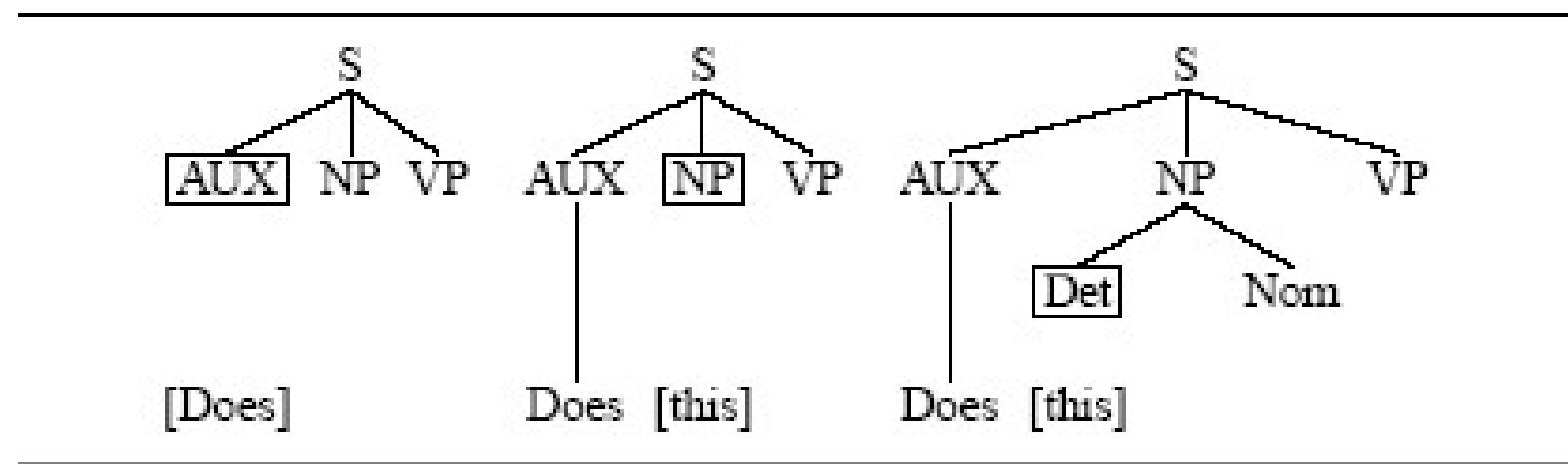
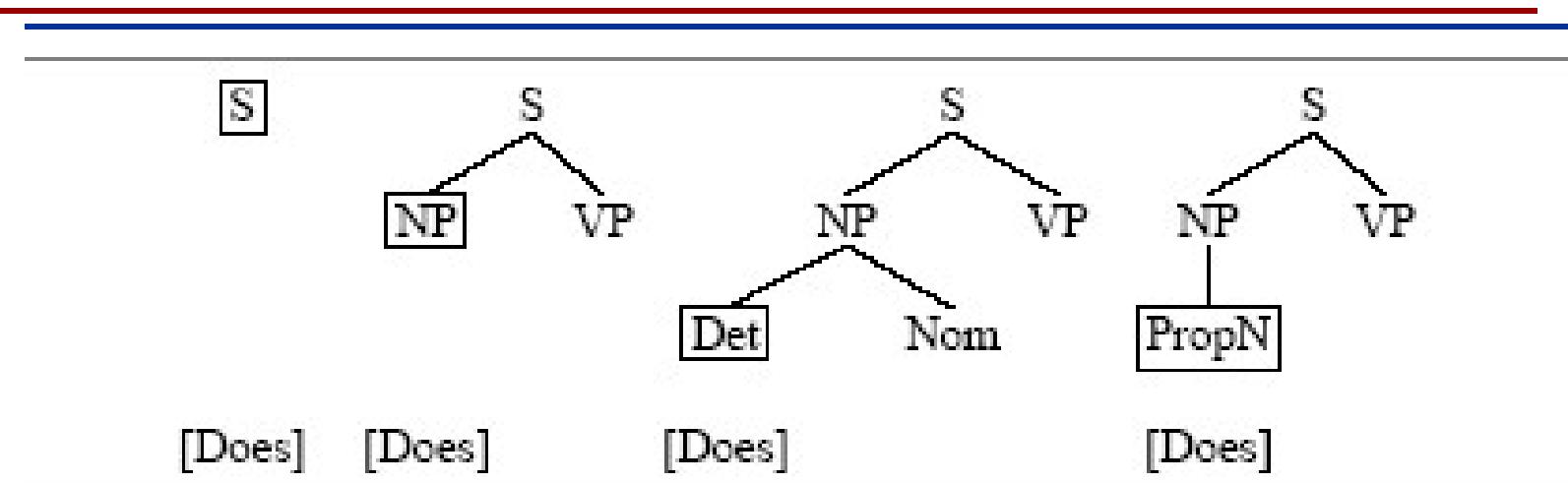
- * Ağacın herhangi bir düğümünün etiketi, w 'nun öneki değilse, bu düğüm ölü bir düğümdür.
- * Türetme işlemi ağacın bir düğümünün etiketi w oluncaya, ya da ağacın tüm yaprakları birer ölü düğüm oluncaya kadar sürdürülür. Etiketi w olan bir düğüm elde edilirse ayrıştırma dumlulu sonuçlanmış olur. Etiketi w olan bir düğüm elde edilemez ve ağacın tüm yapraklarının birer ölü düğüm olduğu anlaşılrsa, ayrıştırma olumsuz sonuçlanmış olur.

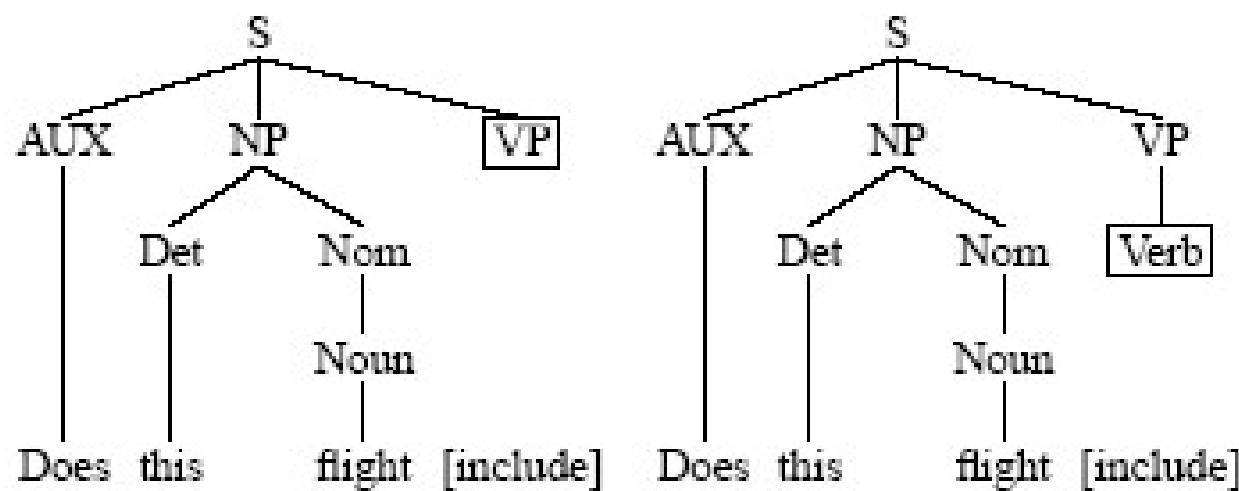
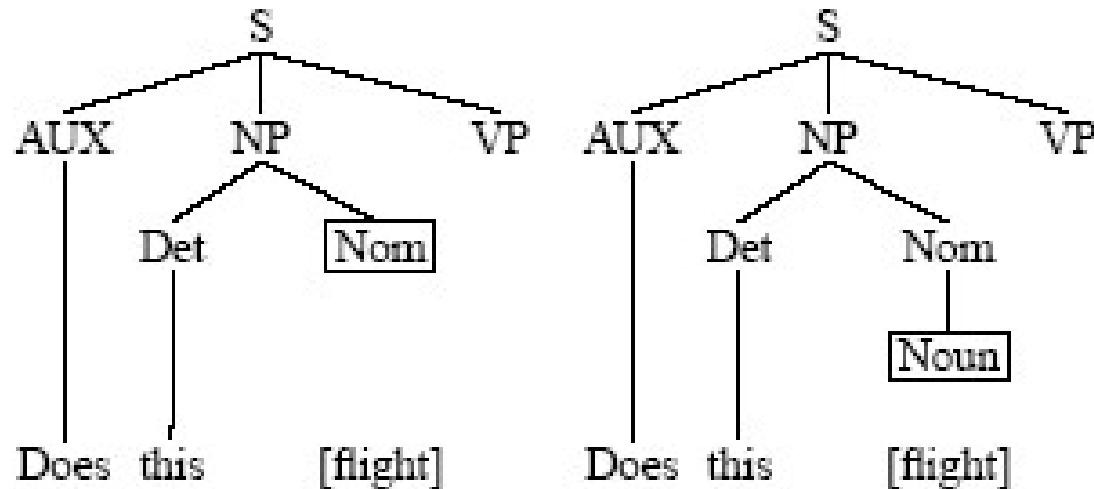


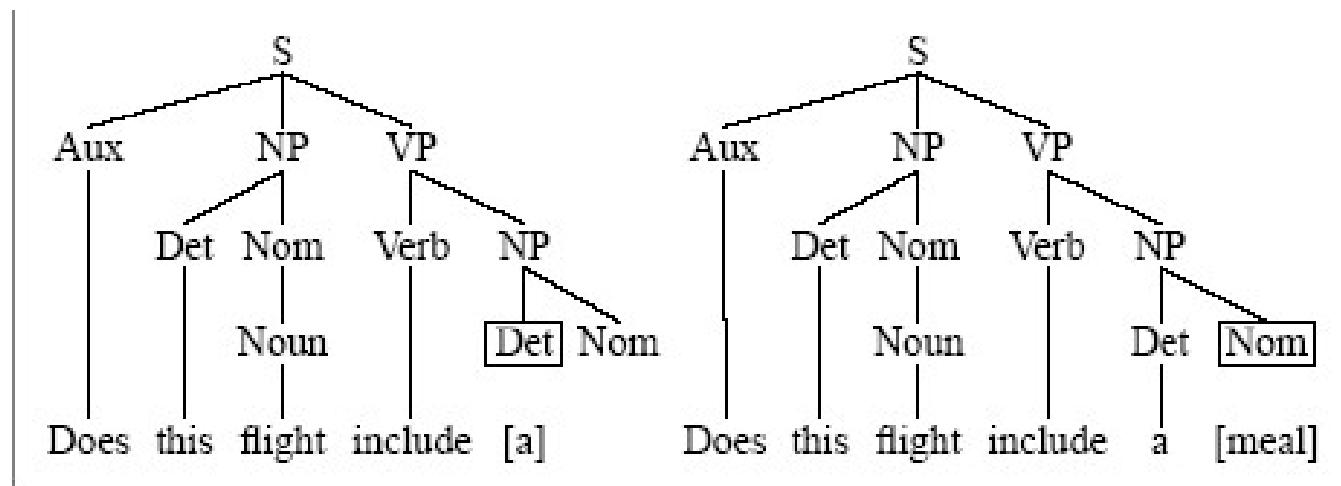
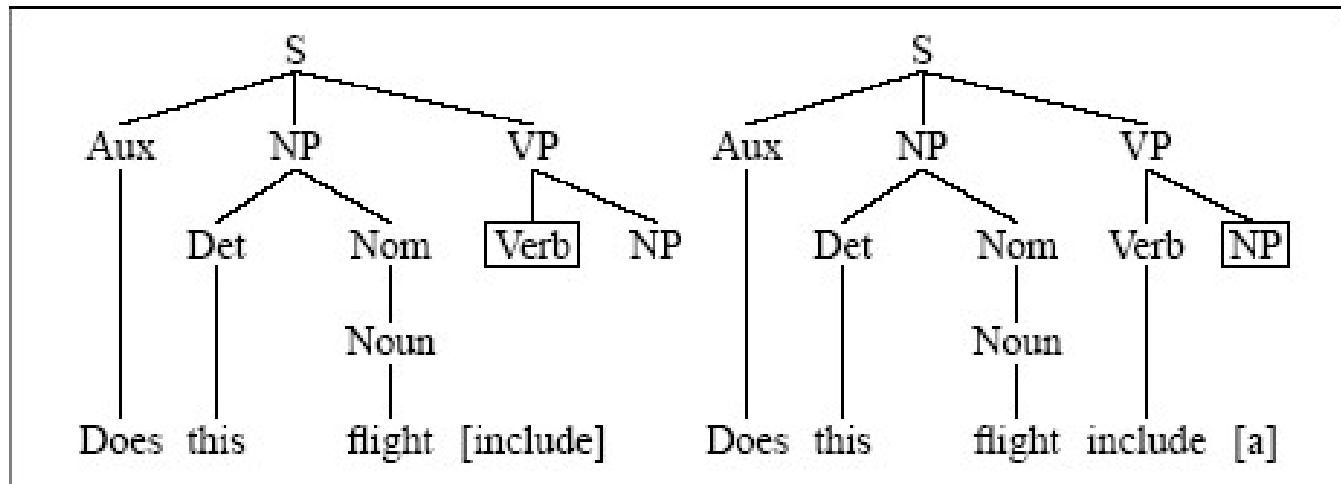
Ayrıştırma işleminin karmaşıklığı, dilbilgisinin yeniden yazma kurallarının sayısı ile ayrıstırılan tümcedeki simge sayısına paralel olarak üssel biçimde artar.

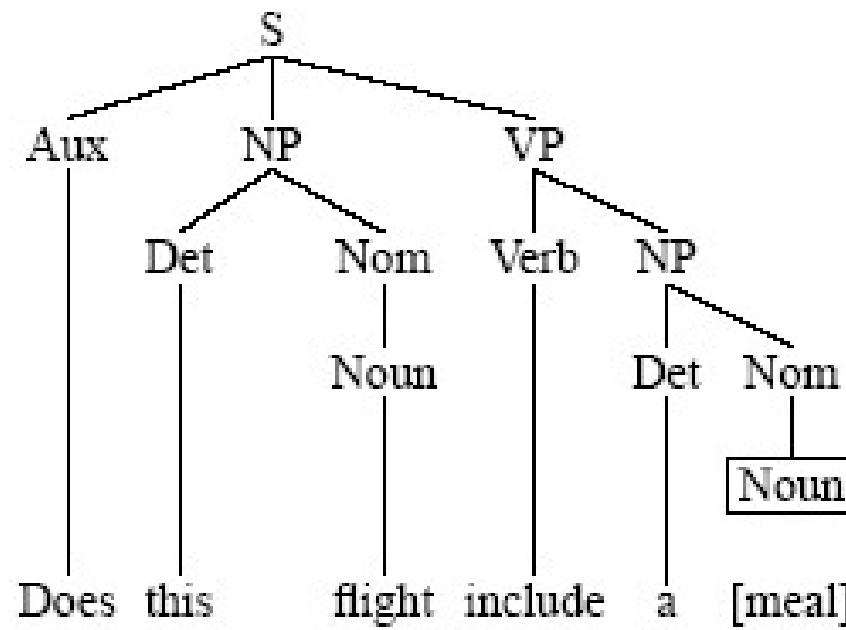
Yukarıdan-Aşağıya Ayrıştırma (Top-Down Parsing, Left-to-Right, Depth-First)

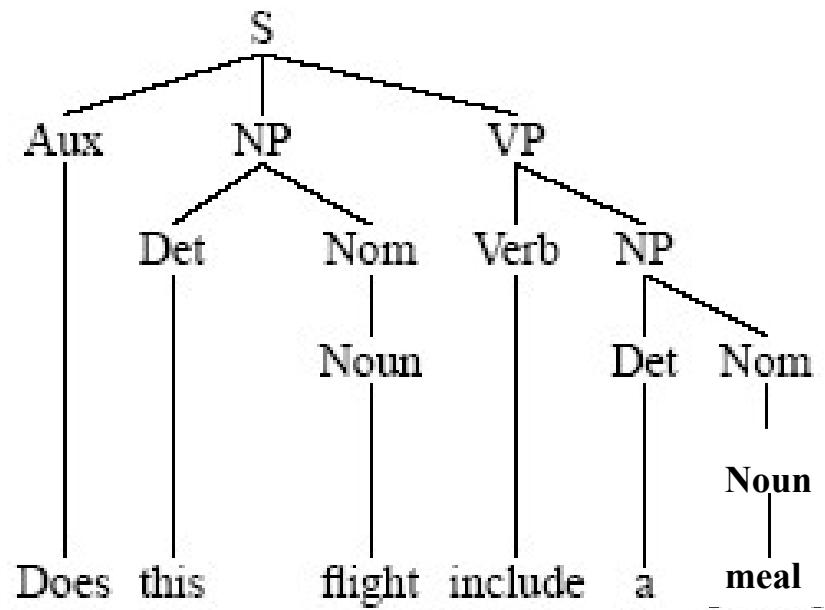




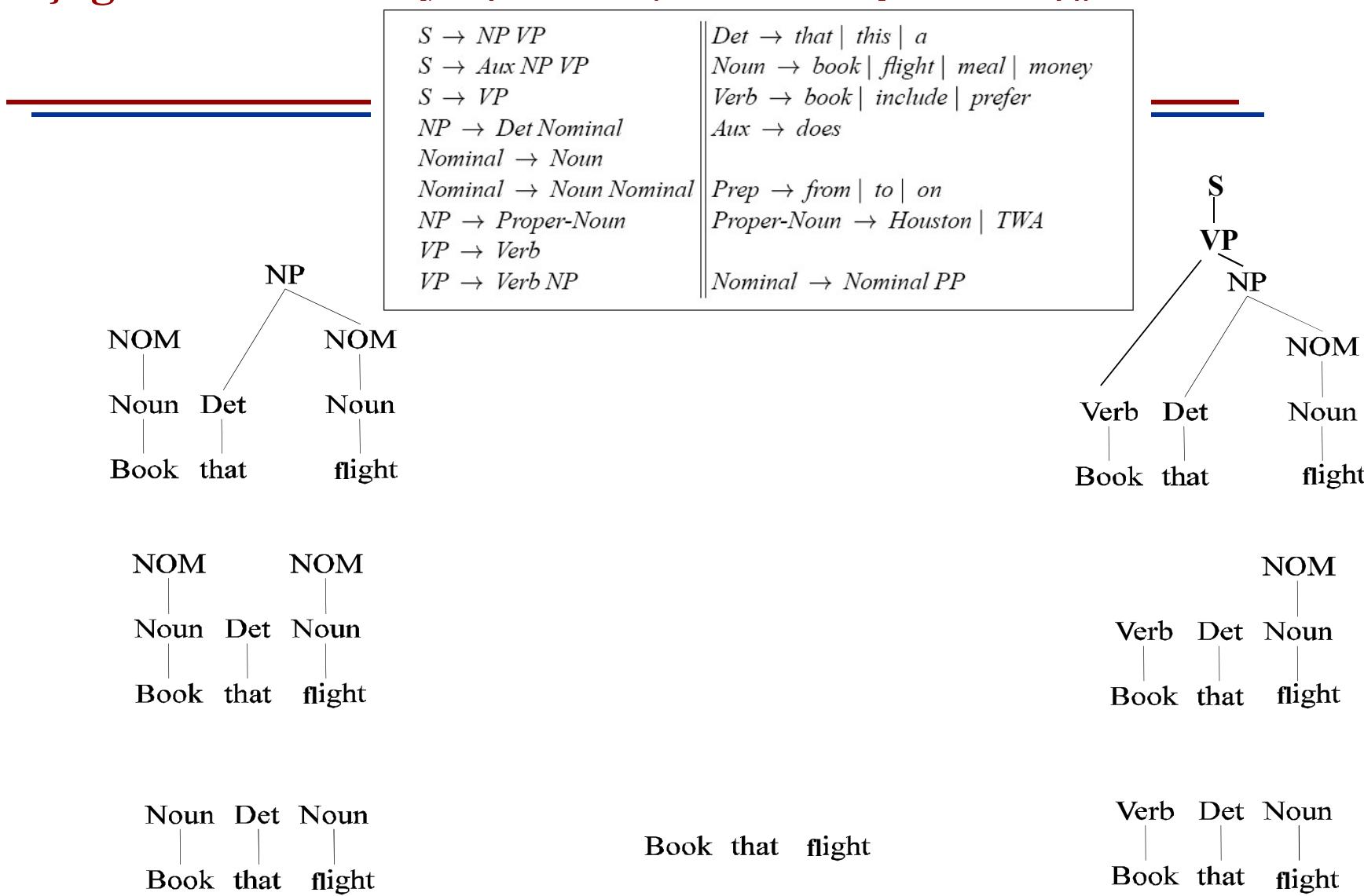








Aşağıdan-Yukarı Ayrıştırma (Bottom-Up Parsing)



S -> Aux NP VP

S -> NP VP

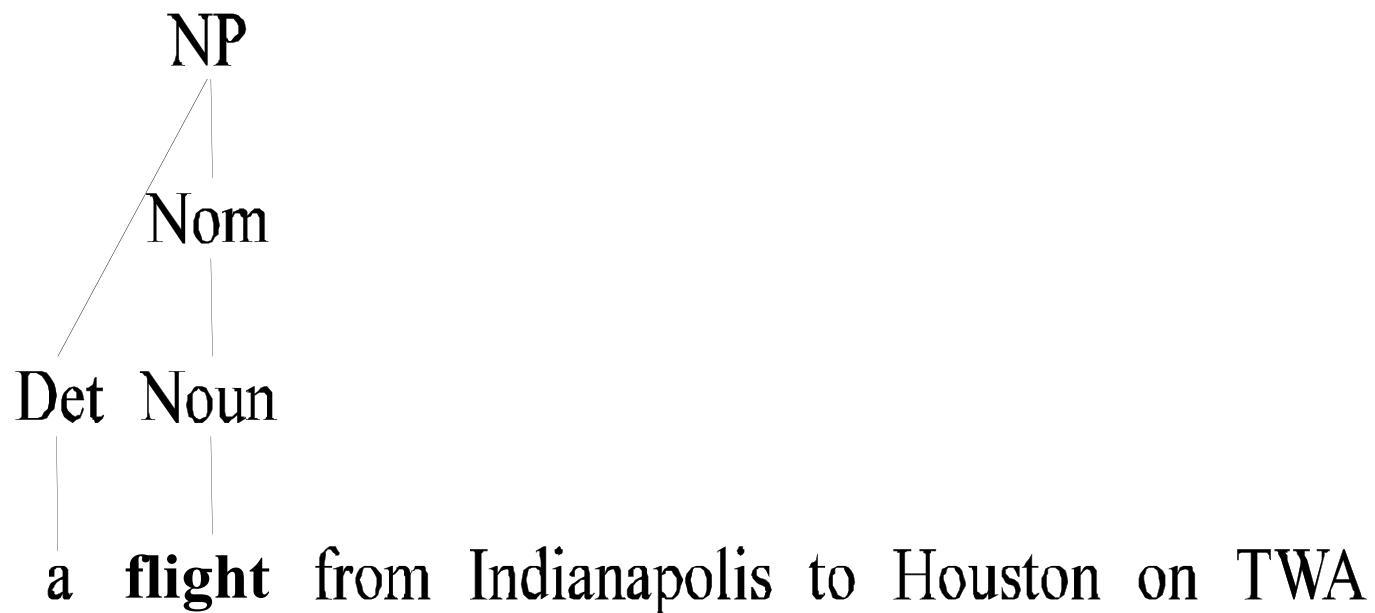
NP -> Det Nominal

NP -> NP PP

A flight from Indianapolis to Houston on TWA

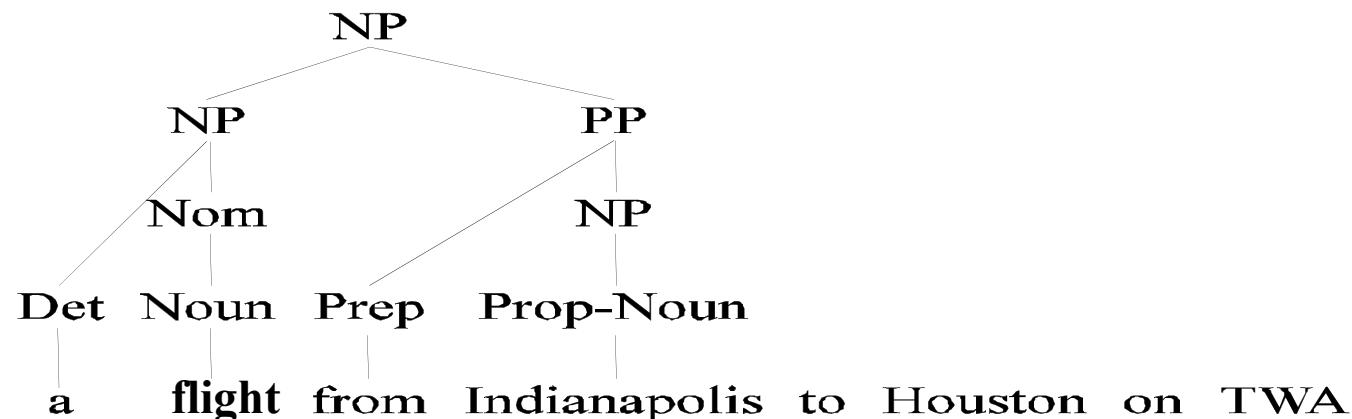
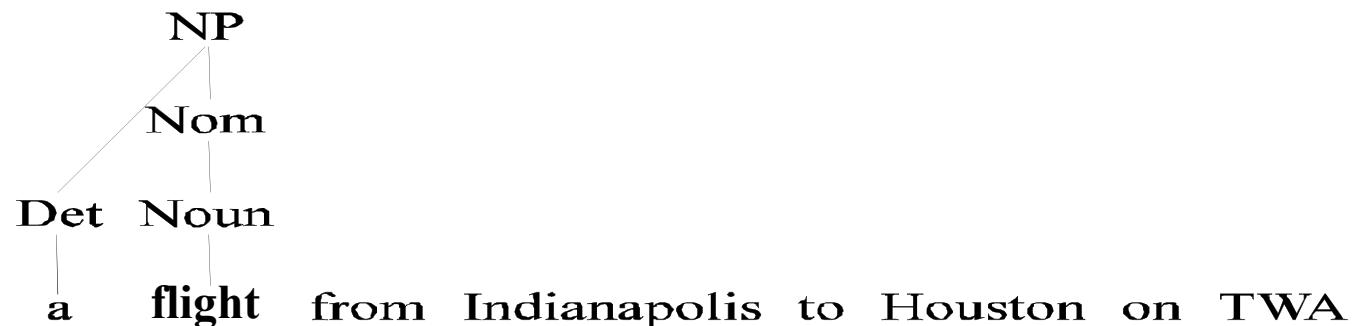
S -> Aux NP VP
S -> NP VP
NP -> Det Nominal
NP -> NP PP

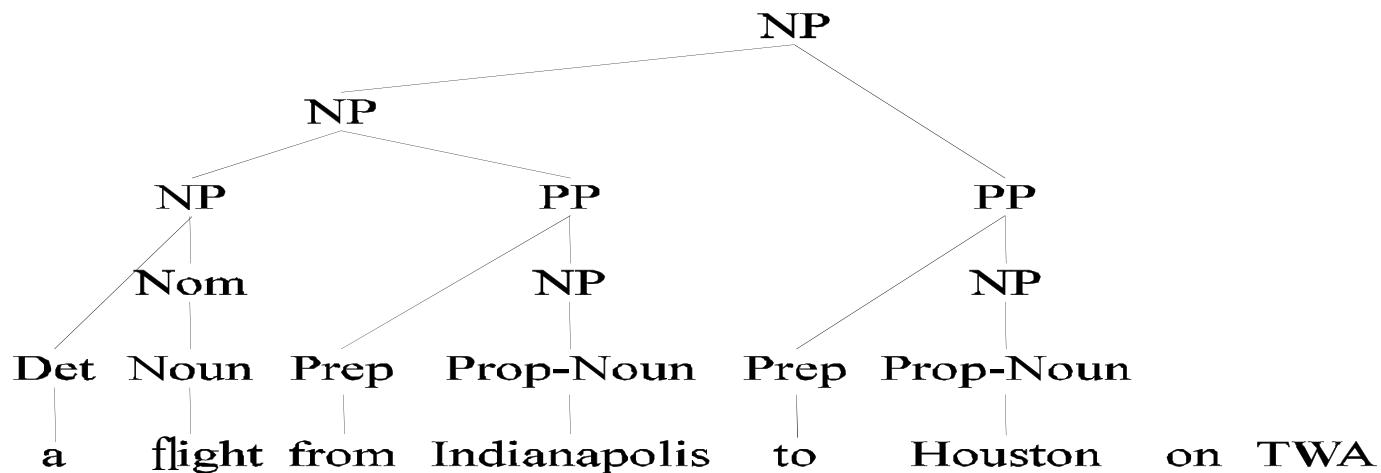
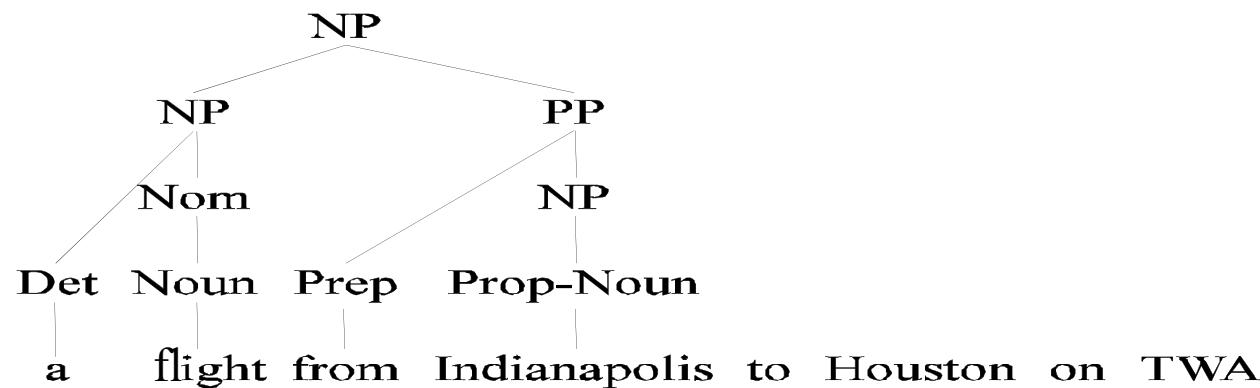
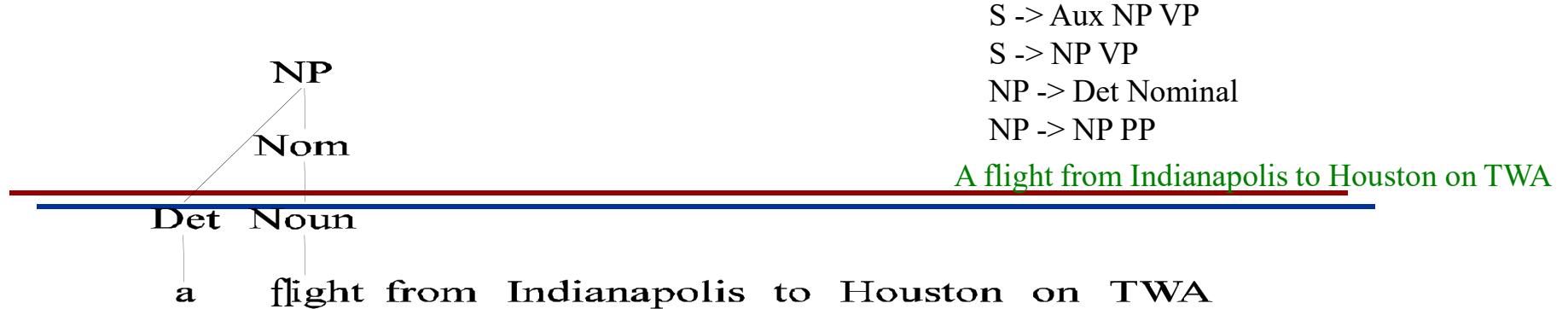
A flight from Indianapolis to Houston on TWA



S -> Aux NP VP
S -> NP VP
NP -> Det Nominal
NP -> NP PP

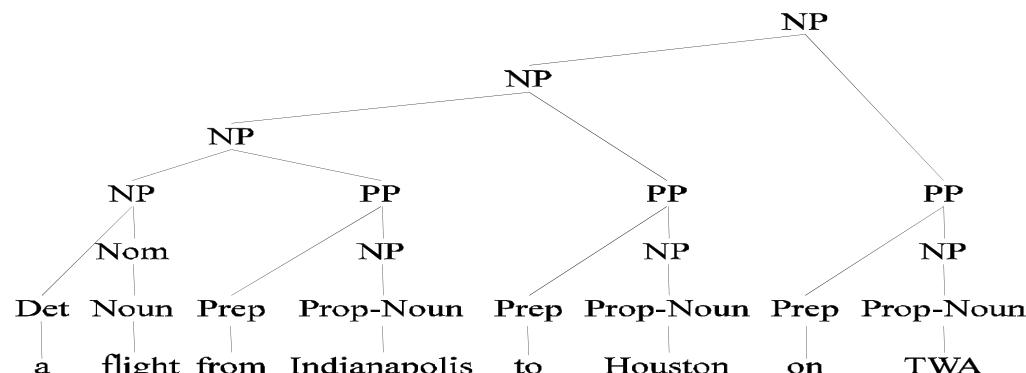
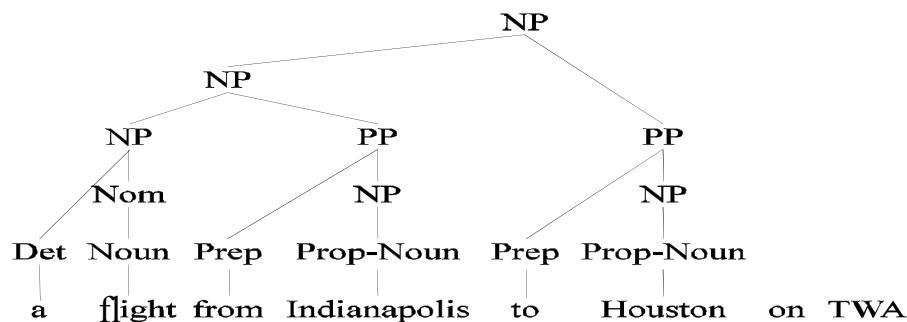
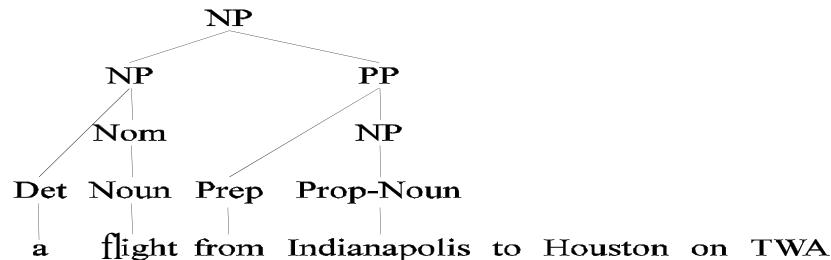
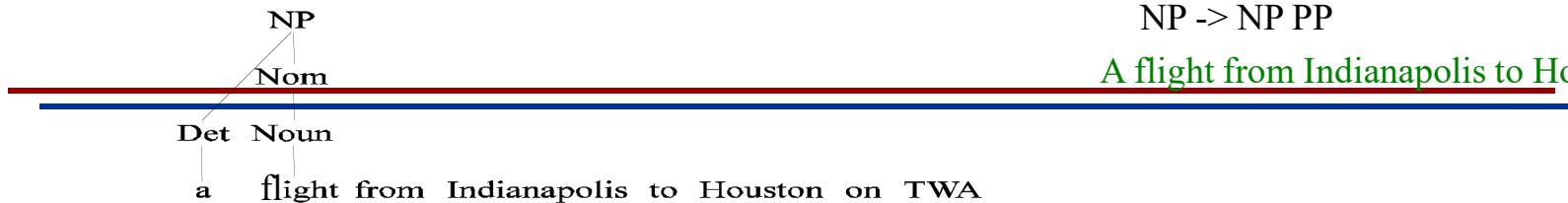
A flight from Indianapolis to Houston on TWA





$S \rightarrow \text{Aux NP VP}$
 $S \rightarrow \text{NP VP}$
 $\text{NP} \rightarrow \text{Det Nominal}$
 $\text{NP} \rightarrow \text{NP PP}$

A flight from Indianapolis to Houston on TWA



Örnek “Bir cümle isim veya fiil grubundan oluşur.”, “isim grubu isim ve/veya sıfat, fiil grubu da isim grubu ve/veya fiil den oluşur”.

Bu gramerde “Küçük çocuk kırmızı top aldı” cümlesinin çözümü

C→İG FG : (cümle isim ve fiil grubundan oluşur)

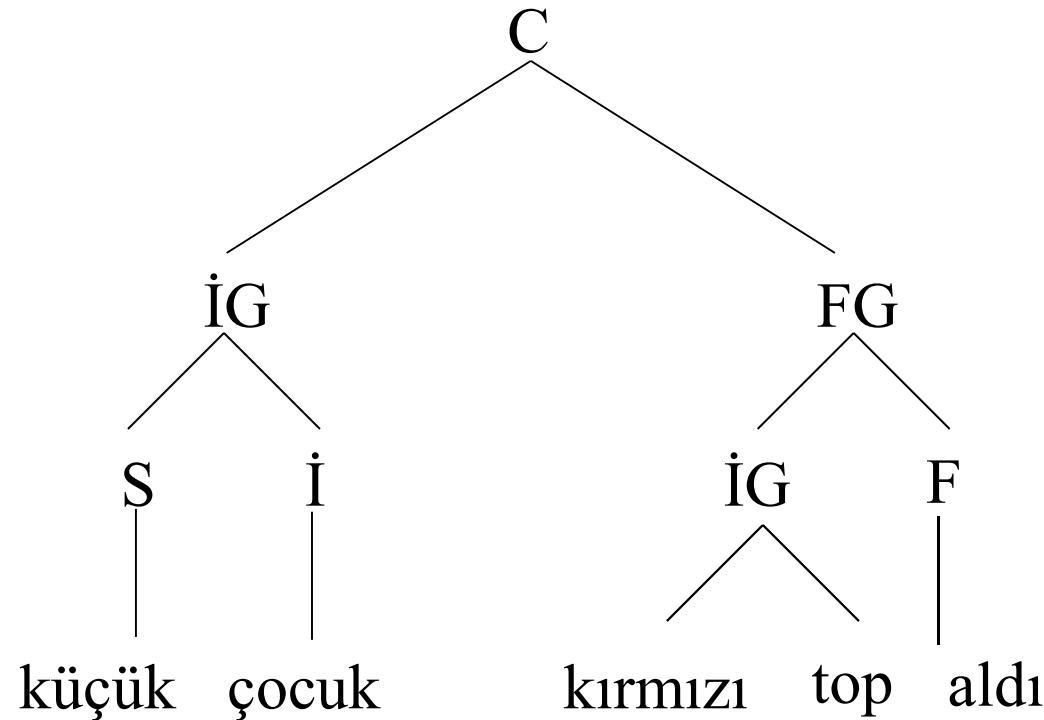
İG→S İ : (isim grubu→sıfat+isim)

FG→İG F : (fiil grubu→isim grubu + fiil)

S→küçük|kırmızı

İ→çocuk|top

F→aldı



Genişletilmiş Geçiş Ağları – GGA

Augmented Network Transition Grammer- ATN

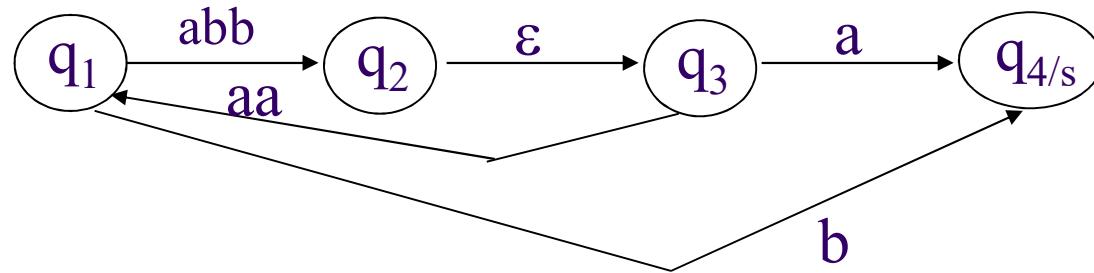
- Biçimsel dillerin, doğal dilleri tam olarak ifade edemediği durumlarda Genişletilmiş Geçiş Ağları tercih edilir.
- GGA, cümlelerin sözdizimsel analizinde kullanılır.
- GGA sonlu makinelere benzeyen durum ve bu durumlar arası geçiş kurallarından oluşmaktadır.

Üç bileşenden oluşur:

- En az bir başlangıç ve son durumu olan sonlu sayıdaki durumlar kümesi
- Belli bir metindeki mümkün olan harflerden oluşan küme
- Sonlu sayıdaki bir durumdan diğer bir duruma geçisi sağlayan geçişler kümesi

-
-
- ❖ Geçiş ağlarında bir durumdan diğer bir duruma geçmek için gerekli harf okunur ve bu harf geçilecek olan duruma geçmek için gereken harfle karşılaştırılır. Uygun olması durumunda geçilir.
 - ❖ Geçiş ağlarında doğru bir yol, bir başlangıç durumundan başlayıp, son duruma ulaşan geçişler sağlandığında tamamlanır.
 - ❖ Geçişlerdeki harflerin birbirine eklenmesiyle oluşan metin, ağın kabul etmesi için verilen metinle aynı ise, bu metin ağ tarafından kabul edilmiş demektir.

Genişletilmiş Geçiş Ağı örneği



Bu ağda tanınabilecek birkaç metin örneği:

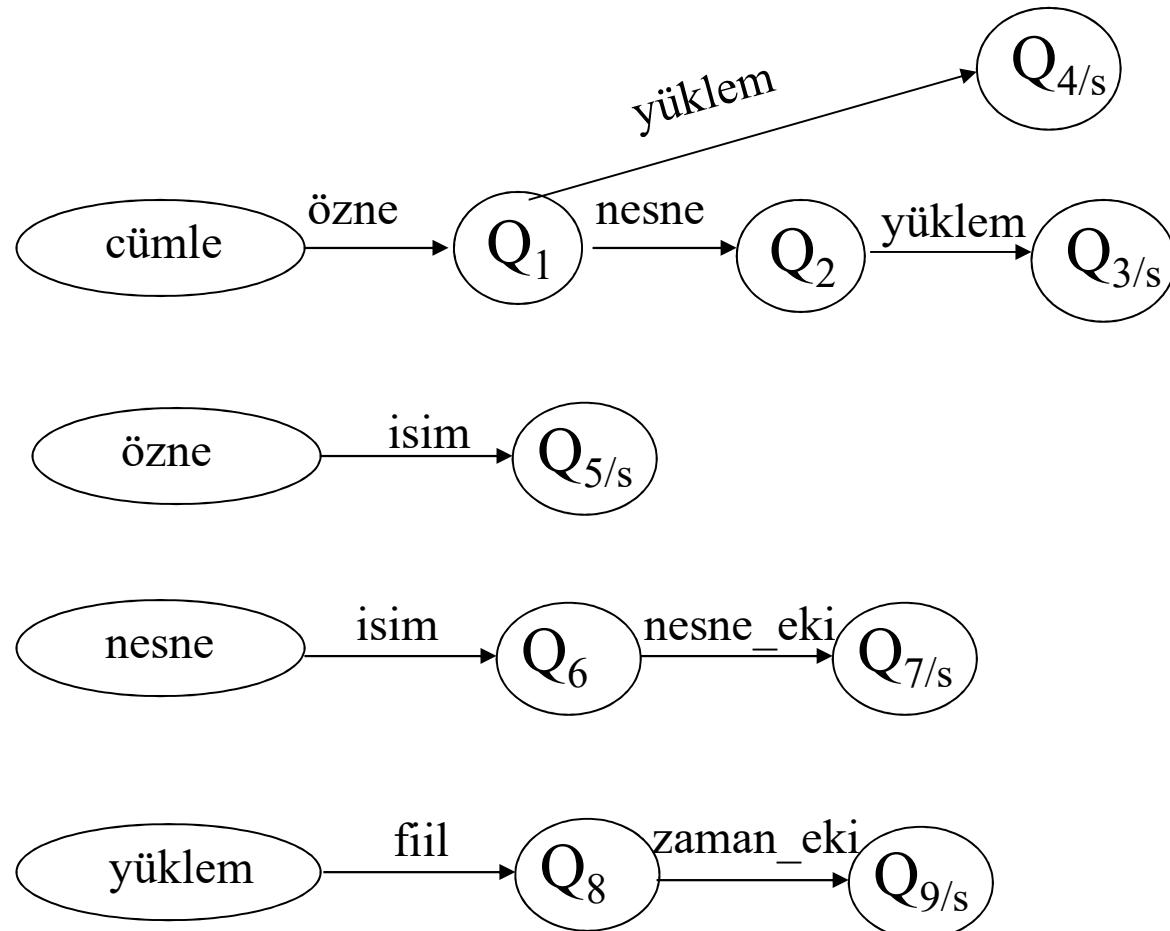
abba, abbaaabba, abbaab, b

Fakat *abbab, baab* yi tanıyamaz.

-
-
- ❖ Dilin grameri, durumlara karşı düşen yönlü graflar şeklinde ifade edilir.
 - ❖ Bağlarda ise, çoğalmakta olan durumlar arasındaki geçişleri tanımlayan etiket sınıfları bulunmaktadır.
 - ❖ Bu etiketler özel kelimeler, kelime kategorileri veya cümlenin önemli parçalarını tanıyan diğer ağlara yönlendirme gibi değerler almaktadır.
 - ❖ GGA, dil tanımında belirtilen cümle birimlerini, hiyerarşik yapı içerisinde yukarıdan aşağıya doğru çözümlemeye çalışır.

Basit bir Türkçe sözdizimi kural kümelerinin GGA yapısı

“Ali camı kırdı” cümlesi morfolojik analiz sonucunda “Ali cam + i kır + di”



-
-
- ❖ Bir sözdizimi kural kümesinin GGA biçiminde ifade edilmesin yararı, cümle birimlerinin bir kez tanımlandıktan sonra sonsuz kez kullanılıyor olmasıdır.
 - ❖ Bir sıfat tamlaması bir kez tanımlanıp birçok yerde kullanılabilir.
 - ❖ GGA'da döngüler bulunabilir. Bu özellik ile dilin kabul ettiği cümle yapıları sayısı sonsuza kadar artırabilir.

“yırtık kırmızı büyük top”

